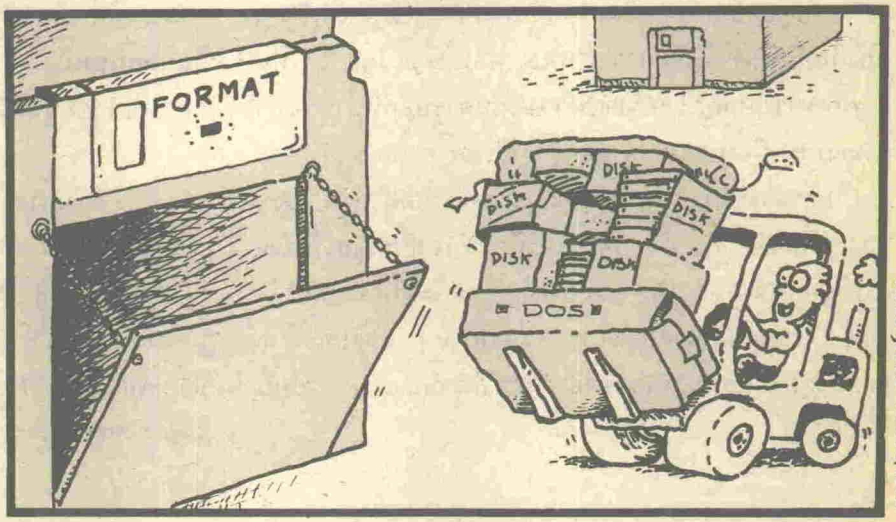
มุมนักเล่น

OS

ชัยยงค์ วงศ์ชัยสุวัฒน์

การฟอร์แมต และการเก็บข้อมูล ในดิสค์ของดอส



ภาพประกอบ : สุพจน์ สุนทร

เสนอรายละเอียดการทำงานของคำสั่ง FORMAT และผลลัพธ์ของการจัดเตรียมดิสค์ของคำสั่งนั้น

แผ่นดิสค์ที่เราซื้อมาใหม่หรือฮาร์ดดิสค์ที่ผ่านการฟอร์แมตระดับต่ำและการจัดพาร์ติชันเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการฟอร์แมตระดับสูงด้วยการใช้คำสั่ง FORMAT ที่เป็นไฟล์ FORMAT.COM มาพร้อมกับโปรแกรมดอส เชื่อว่าพวกเราที่ผ่านการใช้งานไมโครคอมพิวเตอร์ โดยเฉพาะกับเครื่องตระกูลไอบีเอ็มมาแล้ว ต้องรู้หรือเคยใช้คำสั่ง FORMAT มาก่อน มุมโอเอสฉบับนี้จะนำเสนอรายละเอียดการทำงานของคำสั่ง FORMAT และผลลัพธ์ของการจัดเตรียมดิสค์ของคำสั่งนั้น โดยจะกล่าวถึงโครงสร้างทางลอจิคัลของดิสค์และวิธีการจัดเก็บข้อมูลในดิสค์ภายใต้สภาวะแวดล้อมของพีซีดอสหรือเอ็มเอสดอส

ในสมัยก่อนการนำคอมพิวเตอร์มาช่วยงาน ผู้ใช้งานปลายทางที่ไม่ใช่ผู้พัฒนาระบบไม่จำเป็นต้องเรียนรู้โครงสร้างการจัดเก็บข้อมูลในดิสค์เลย แต่ในสถานภาพปัจจุบันหากเราสามารถเรียนรู้และเข้าใจวิธีการจัดเก็บข้อมูลในดิสค์ ก็จะเป็นผลดีทำให้เราสามารถใช้งานคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ ช่วยในการบำรุงรักษาข้อมูลที่เก็บภายในดิสค์ และสุดท้ายให้เข้าใจแนวการป้องกันและกู้ข้อมูลคืนอันเนื่องมาจากการรับไวรัสคอมพิวเตอร์ที่กำลังระบาดในปัจจุบัน

ดอสใช้ดิสค์เก็บข้อมูลอย่างไร

แผ่นดิสเกตต์ที่ซื้อมาใหม่ ๆ เราจะนำมาใช้งานเลยทันทีไม่ได้ต้องมาทำการฟอร์แมตก่อน และแผ่นดิสค์นี้อาจนำไปใช้กับเครื่องต่างรุ่นต่างตระกูลได้ บนเครื่องไมโครคอมพิวเตอร์ตระกูลไอบีเอ็มที่ใช้โปรแกรมควบคุมดำเนินงานเป็นระบบเอ็มเอสดอสจะเรียกไฟล์ FORMAT. COM มาทำงานเพื่อฟอร์แมตแผ่นดิสค์

ดิสเกตต์ที่นำมาใช้งานปัจจุบันจัดเป็นแบบ Soft Sector คือจะเป็นหน้าที่ของซอฟต์แวร์คือโปรแกรม FORMAT นี้เองที่จะทำการจัดแบ่งพื้นที่บนผิวจานแม่เหล็กออกเป็นวง ๆ เรียกแทร็ก (track) และแบ่งเป็นท่อน ๆ เรียกเซกเตอร์ (Sector) และจะบันทึกข้อมูลดัมมี่ลงสู่ทุกเซกเตอร์ไว้ก่อน คำสั่ง FORMAT จะตรวจสำรวจทุกเซกเตอร์ไปทีละแทร็กและขณะเดียวกันจะบันทึกข้อมูลพิเศษเรียก timing mark จำนวนสี่ไบต์ลงที่เซกเตอร์ต่าง ๆ เพื่อใช้อ้างอิงบอกตำแหน่งของเซกเตอร์

คำสั่ง FORMAT จะทำการตรวจสอบดิสค์ว่ามีแทร็กใดเสียบ้าง ถ้าแทร็กใดเสียก็จะมีการหมายป้องกันไม่ให้นำไปใช้บันทึกข้อมูล การตรวจสอบและบันทึกข้อมูลจะทำงานเป็น

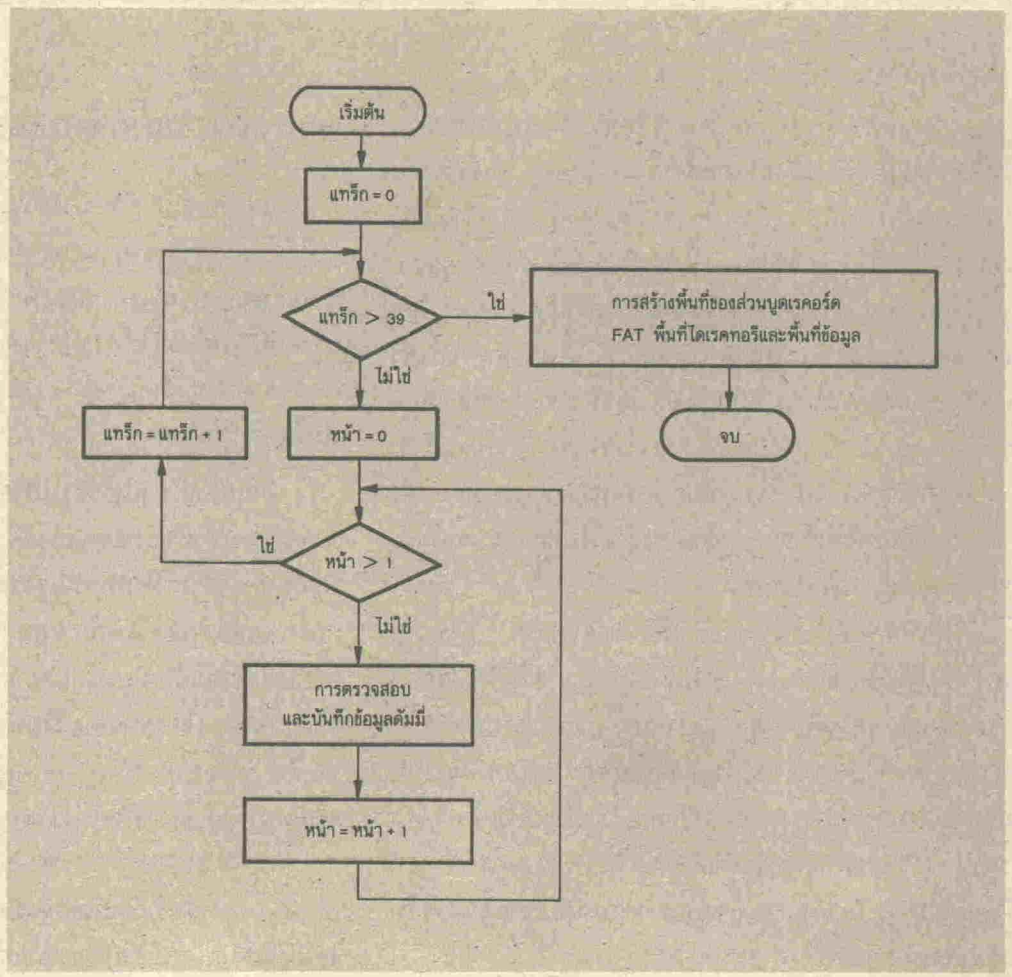
ลำดับ นับจากแทร็กวงนอกสุดที่เป็นแทร็ก 0 ทั้งหน้า 0 และหน้า 1 เข้าไปด้านในจนถึงแทร็กที่ 39 (ถ้ามี 40 แทร็ก กรณีเป็นแผ่น 5.25 นิ้ว แบบ 48 TPI) หรือแทร็กที่ 79 (ถ้ามี 80 แทร็ก กรณีเป็นแผ่น 5.25 นิ้วแบบ 96 TPI หรือขนาด 3.5 นิ้ว แบบ 96 TPI) รูปที่ 1 เป็นผังลำดับการทำงานของคำสั่ง FORMAT สำหรับแผ่นดิสค์ที่มีจำนวนแทร็กเพียง 40 แทร็ก

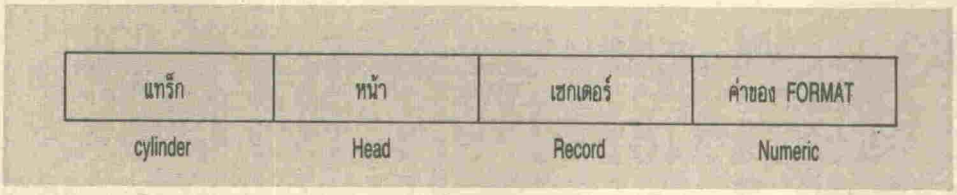
ข้อมูลพิเศษจำนวน 4 ไบต์เป็นตัวแสดงเซกเตอร์ต่าง ๆ ของดิสค์ ซึ่งเราจะเรียกว่า CHRN ดังรูปที่ 2 โดย C เป็นการแทนไซเลนเดอร์ (Cylinder) H เป็นการบอกหน้าหรือหัว (Head) R เป็นการแทนเซกเตอร์หรือเรคอร์ด (Record) และ N เป็นค่าตัวเลขบอกขนาดไบต์ของการฟอร์แมต ค่า CHRN ของเซกเตอร์ต่าง ๆ ตั้งแต่เซกเตอร์ที่ 1 จนถึงเซกเตอร์ที่ 720 สำหรับแผ่นขนาดความจุ 360 กิโลไบต์ จะเรียงในเซกเตอร์ดังรูปที่ 3 โดยตำแหน่งเซกเตอร์แรกของหน้า 0 และแทร็ก 0 ระบบจะตรวจสอบรู้จากช่องอินเด็กซ์ที่เรามองเห็นจากแผ่นได้เลย

ค่าตัวเลข N เป็นตัวบอกขนาดของเซกเตอร์ โดยถ้า N = 0, 1, 2 หรือ 3 จะแทนขนาดเซกเตอร์เป็น 128, 256, 512 หรือ 1024 ไบต์ตามลำดับ ตามมาตรฐานของไอบีเอ็มหรือเอ็มเอสดอส จะใช้ค่า N = 2 เสมอคือยึดขนาดเซกเตอร์มาตรฐานเป็น 512 ไบต์เสมอ (เราอาจเปลี่ยนค่าของ N เป็นค่าตัวเลขอื่นที่มีขนาดเซกเตอร์ไม่เป็นมาตรฐานด้วยการเรียกผ่านทางโปรแกรมไบออส กรณีนี้เป็นวิธีการป้องกันก๊อบปี้ข้อมูลแบบหนึ่งเพราะโปรแกรมดอสเพื่อการคัดลอกปกติจะมองไม่เห็นข้อมูลที่ผิดมาตรฐานนี้)

หลังจากตรวจสอบและเติมค่า CHRN กับข้อมูลดัมมี่ลงสู่เซกเตอร์ต่าง ๆ จนครบทุกหน้าทุกแทร็กเรียบร้อย โปรแกรม FORMAT จะสร้างพื้นที่สำคัญ 4 ส่วนบนแผ่นดิสเกตต์คือ บูตเรคอร์ด FAT หรือผังการจัดวางไฟล์บนดิสค์ที่ย่อมาจาก File Allocation Table พื้นที่ไดเรคทอรี และพื้นที่เก็บข้อมูลไฟล์ รายละเอียดของพื้นที่ทั้งสี่จะกล่าวถึงในตอนต่อไป

รูปที่ 1
การทำงานของคำสั่ง
FORMAT





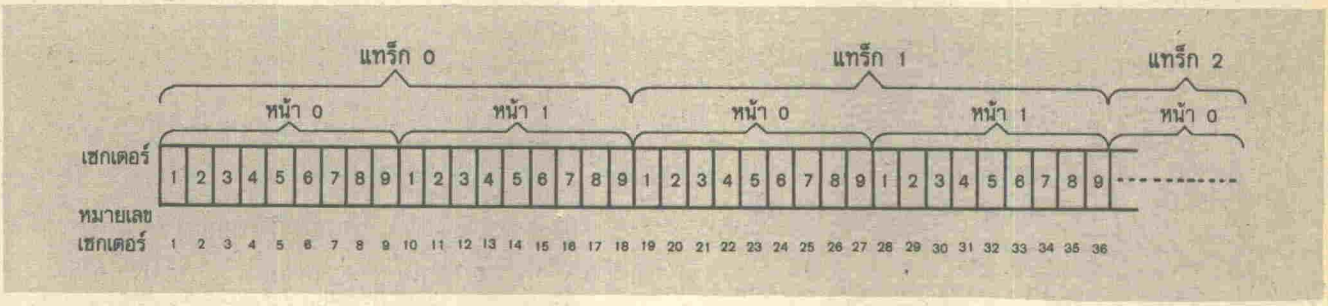
รูปที่ 2
ข้อมูล timing mark ที่บันทึกค่า
จำนวน 4 ไบต์ลงที่ทุก ๆ เซกเตอร์

รูปที่ 3
ค่า CHRN จำนวน 4 ไบต์
ที่อยู่ในเซกเตอร์ต่าง ๆ

ดิสค์ตามมาตรฐานของไอบีเอ็ม	ขนาด(นิ้ว)	หน้า	แทร็ก	เซกเตอร์ต่อแทร็ก	ขนาดเซกเตอร์	ขนาดความจุดิสค์ (กิโลไบต์)
แผ่นหน้าเดียวความจุเท่าเดียว	5.25	1	40	8	512	160
แผ่นสองหน้าความจุเท่าเดียว	5.25	2	40	8	512	320
แผ่นหน้าเดียวความจุสองเท่า	5.25	1	40	9	512	180
แผ่นสองหน้าความจุสองเท่า	5.25	2	40	9	512	360
แผ่นสองหน้าความจุหนาแน่นสูง	5.25	2	80	15	512	1200
แผ่นสองหน้าความจุปกติ	3.5	2	80	9	512	720
แผ่นสองหน้าความจุหนาแน่นสูง	3.5	2	80	18	512	1440
ฮาร์ดดิสค์สำหรับ XT	5.25	4	305	17	512	10370
ฮาร์ดดิสค์สำหรับ AT	5.25	4	614	17	512	20876

รูปที่ 4
ขนาดความจุของดิสเกตต์และ
ฮาร์ดดิสค์ขนาดต่าง ๆ

รูปที่ 5
ดอสนับหมายเลขเซกเตอร์
เรียงจากแทร็ก 0 วงนอกสุด
เข้าไปสู่แทร็กวงในสุด ▼



การแบ่งจำนวนแทร็กและเซกเตอร์

คำสั่ง FORMAT ของดอสเวอร์ชันแรกจะฟอร์แมตแผ่นดิสค์มีความหนาแน่นน้อย คือ จำนวน 8 เซกเตอร์ต่อแทร็ก แต่ในเวอร์ชันหลัง ๆ ก็จะฟอร์แมตแผ่นดิสค์ขนาด 5.25 นิ้ว ให้หนาแน่นมากขึ้นเป็น 9 เซกเตอร์ต่อแทร็กสำหรับกรณีเป็นแผ่นแบบความหนาแน่นสองเท่า หรือเป็น 15 เซกเตอร์ต่อแทร็กสำหรับแผ่นความหนาแน่นสูง กรณีแผ่นขนาด 3.5 นิ้วจำนวนเซกเตอร์ต่อแทร็กขยายขึ้นเป็น 9 หรือ 18 สำหรับความหนาแน่นปกติหรือความหนาแน่นสูง

ในปัจจุบันแผ่นดิสเกตต์และฮาร์ดดิสค์ขนาดต่าง ๆ จะมีการแบ่งจำนวนแทร็กและเซกเตอร์ตามมาตรฐานไอบีเอ็มหรือเอ็มเอสดอส ดังรูปที่ 4 และการคำนวณขนาดความจุของดิสค์จะเกิดจากการนำจำนวนหน้า จำนวนแทร็กหรือไซเลนเดอร์ จำนวนเซกเตอร์ต่อแทร็กและขนาดเซกเตอร์มาคูณกัน

ในการเรียงลำดับพื้นที่บนแผ่นดิสค์ ดอสจะมองเรียงจากเซกเตอร์ 1 ไปจนถึงเซกเตอร์สุดท้ายที่มีอยู่ ลักษณะการนับเซกเตอร์จะเริ่มจากแทร็กวงนอกสุด โดยนับหน้า 0 แทร็ก 0 ไปทีละเซกเตอร์จนถึงเซกเตอร์สุดท้ายของแทร็กแล้วไปเริ่มหน้า 1 แทร็ก 0 ที่อยู่คนละหน้าของจาน นับหมายเลขเซกเตอร์ต่อจากหน้าแรกจนถึงเซกเตอร์สุดท้ายของแทร็ก จากนั้นก็เริ่มในแทร็ก 1 ที่อยู่ถัดไปในหน้า 0 และหน้า 1 ตามลำดับ

รูปที่ 6
ข้อมูลในบูตเรคอร์ด
ของแผ่นดิสค์ขนาด
360 กิโลไบต์

```
00  EB 28 90 49 42 4D 20 20 33 2E 33 00 02 02 01 00   δ(ÉIBM 3.3.....
10  02 70 00 D0 02 FD 02 00 09 00 02 00 00 00 00 00   .p.╨.².........
20  00 00 00 00 00 00 00 00 00 00 FA 33 ED B8 C0 07   ..........3φ┐└.
30  8E D8 C4 1E 1C 00 88 16 FD 01 0A D2 79 08 89 1E   Ä╪...ê.².╥y.ë.
40  24 00 8C 06 26 00 8E C5 8E D5 BC 00 7C FC 1E 36   $.î.&.Ä╪Ä╒╝.|ⁿ.6
50  C5 36 78 00 BF 2A 7C B9 0B 00 F3 A4 1F C6 06 2E   ┼6x.┐*|╣..≤ñ.├..
60  00 0F BF 78 00 B8 2A 7C AB 91 AB FB 8A 16 FD 01   ..┐x.┐*|½æ½√è.².
70  CD 13 A0 10 00 98 F7 26 16 00 03 06 0E 00 E8 73   =.á..ÿ≈&......Φs
80  00 E8 79 00 BB 00 05 53 E8 A0 00 5F BE 71 01 B9   .Φy.╗..SΦá. ╛q.╣
90  0B 00 90 F3 A6 75 57 83 C7 15 B1 0B 90 90 F3 A6   ..É≤ªuWâ╟.▒.▓.É≤ª
A0  75 4C 26 8B 47 1C 99 8B 0E 0B 00 03 C1 48 F7 F1   uL&ïG.Öï....┴H≈±
B0  3D 14 00 7F 02 B0 14 96 A1 11 00 B1 04 D3 E8 E8   =..⌂.▒.ûí..▒.└ΦΦ
C0  32 00 FF 36 24 00 C4 1E 6D 01 E8 30 00 E8 5B 00   2. 6$.─.m.Φ0.Φ[.
D0  2B F0 76 0D E8 1D 00 52 F7 26 0B 00 03 D8 5A EB   +≡v.Φ..R≈&...╪Zδ
E0  E9 5B 8A 2E 15 00 8A 16 FD 01 FF 2E 6D 01 BE 8B   Θ[è...è.². .m.╛ï
F0  01 EB 54 90 01 06 24 00 11 2E 26 00 C3 A1 18 00   .δTÉ..$...&.├í..
100 F6 26 1A 00 91 A1 24 00 8B 16 26 00 F7 F1 92 8B   ÷&..æí$.ï.&.≈±Æï
110 0E 18 00 F6 F1 2A CC 91 FE C5 86 E9 D0 CE D0 CE   ...÷±*╠æ■┼åΘ╨╬╨╬
120 0A F1 86 F2 87 CA 8A 16 FD 01 C3 BF 05 00 B8 01   .±å≥ç╩è.².├┐..┐.
130 02 CD 13 72 03 B0 01 C3 80 FC 11 74 F8 33 C0 CD   .=.r.▒.├Çⁿ.t°3└═
140 13 4F 75 EA BE D5 01 E8 1D 00 BE AB 01 E8 17 00   .OuΩ╛ ╒.Φ..╛½.Φ..
150 33 C0 CD 16 36 C7 06 72 04 34 12 EA 00 00 FF FF   3└═.6╟.r.4.Ω..
160 B4 0E BB 07 00 CD 10 AC 3C 24 75 F4 C3 00 00 70   ┤.╗..═.¼<$u⌠├..p
170 00 49 42 4D 42 49 4F 20 20 43 4F 4D 49 42 4D 44   .IBMBIO  COMIBMD
180 4F 53 20 20 43 4F 4D 00 00 00 00 0A 0D 4E 6F 6E   OS  COM......Non
190 2D 53 79 73 74 65 6D 20 64 69 73 6B 20 6F 72 20   -System disk or
1A0 64 69 73 6B 20 65 72 72 6F 72 24 0A 0D 52 65 70   disk error$..Rep
1B0 6C 61 63 65 20 61 6E 64 20 73 74 72 69 6B 65 20   lace and strike
1C0 61 6E 79 20 6B 65 79 20 77 68 65 6E 20 72 65 61   any key when rea
1D0 64 79 0A 0D 24 0A 0D 44 69 73 6B 20 62 6F 6F 74   dy..$..Disk boot
1E0 20 66 61 69 6C 75 72 65 24 43 6F 70 72 2E 20 43    failure$Copr. C
1F0 4F 4D 50 41 51 20 31 39 38 33 2D 38 37 00 55 AA   OMPAQ 1983-87.U¬
```

รูปที่ 7
รายการข้อมูลในบูตเรคอร์ด
แสดงส่วนที่เป็นพารามิเตอร์ไบออส

ไบต์ที่	ความยาว (ไบต์)	คำอธิบาย
0-2	3	คำสั่งกระโดดของโปรเซสเซอร์
3-10	8	ชื่อและหมายเลขเวอร์ชันของดอส
11-12	2	จำนวนไบต์ต่อเซกเตอร์
13	1	จำนวนเซกเตอร์ต่อคลัสเตอร์
14-15	2	จำนวนเซกเตอร์ที่สงวนเป็นบูตเรคอร์ด
16	1	จำนวนชุดของ FAT
17-18	2	จำนวนรายการสูงสุดที่บรรจุในไดเรคทอรีราก
19-20	2	จำนวนเซกเตอร์ทั้งหมดบนดิสค์
21	1	รหัสประเภทของดิสค์
22-23	2	จำนวนเซกเตอร์ต่อ FAT หนึ่งชุด
24-25	2	จำนวนเซกเตอร์ต่อแทร็ก
26-27	2	จำนวนหน้าหรือหัวอ่านเขียน
28-29	2	จำนวนเซกเตอร์ที่สงวนเฉพาะ

เรานับในลักษณะนี้เข้าไปเรื่อย ๆ จนถึงแทร็กในที่สุด รูปที่ 5 เป็นตัวอย่างการนับเซกเตอร์ของแผ่นดิสค์แบบสองหน้าและความหนาแน่น 9 เซกเตอร์ต่อแทร็ก

กรณีฮาร์ดดิสค์หมายเลขเซกเตอร์จะเรียงจากไซเลนเดอร์นอกสุดคือ ไซเลนเดอร์ 0 เริ่มจากเซกเตอร์ต่าง ๆ ในหน้า 0 แล้วไปหน้า 1, 2 และ 3 (กรณีมี 2 แผ่นรวม 4 หน้า) แล้วจึงขยับเข้าไซเลนเดอร์ 1 ในลักษณะเช่นนี้จนถึงไซเลนเดอร์สุดท้ายที่มีอยู่

บูตเรคอร์ด

บูตเรคอร์ดเป็นพื้นที่ที่ดอสจัดให้อยู่ที่เซกเตอร์แรกคือ แทร็ก 0 หน้า 0 เซกเตอร์ 1 ของแผ่นดิสเกตต์เสมอ โดยจะบรรจุโปรแกรมภาษาแอสเซมบลีสั้น ๆ ที่ถูกเรียกใช้เมื่อเราบูตระบบจากดิสเกตต์ทุกครั้ง โปรแกรมนี้จะสั่งให้ไบออสไปมองหาไฟล์ระบบของดอส (ในที่นี้คือไฟล์ IBMBIO.COM หรือ IO.SYS และ IBMDOS.COM หรือ MSDOS.SYS) ที่บรรจุในดิสค์เพื่อโหลดลงสู่หน่วยความจำและส่งการควบคุมไปยังโปรแกรมระบบนั้น

นอกจากโปรแกรมภาษาแอสเซมบลีเล็ก ๆ นั้นแล้ว ในบูตเรคอร์ดจะใช้เก็บข้อมูลเกี่ยวกับเวอร์ชันของดอส ข้อมูลบอกการฟอร์แมตดิสค์ และข่าวสารเพื่อบอกข้อผิดพลาดพื้นฐานถ้าเกิดการใช้งานผิด ตัวอย่างข้อมูลในบูตเรคอร์ดของแผ่นดิสค์ขนาด 360 กิโลไบต์แสดงได้ดังรูปที่ 6 โดยมีรายละเอียดของแต่ละไบต์ดังรูปที่ 7 และรหัสประเภทดิสค์พร้อมพารามิเตอร์ดิสค์ได้แสดงดังรูปที่ 8

(ข้อมูลที่เหมือนกันทุกประเภทคือ จำนวนไบต์ต่อเซกเตอร์เป็น 512 ไบต์
จำนวนชุดของ FAT เป็น 2 และจำนวนเซกเตอร์ของบูตเรคอร์ดเป็น 1)

รหัสประเภทดิสค์ (ไบต์ที่ 21)	ขนาดและความจุ	เซกเตอร์ต่อแทร็ก (ไบต์ที่ 24-25)	จำนวนหน้า (ไบต์ที่ 26-27)	จำนวนเซกเตอร์ทั้งหมด (ไบต์ที่ 19-20)	จำนวนเซกเตอร์ของ FAT หนึ่งชุด (ไบต์ที่ 22-23)	จำนวนไฟล์ในไดเรคทอรีราก (ไบต์ที่ 17-18)	จำนวนเซกเตอร์ต่อคลัสเตอร์ (ไบต์ที่ 13)
FC	แผ่น 5.25 นิ้ว 160 KB	8	1	320	1	64	1
FD	แผ่น 5.25 นิ้ว 320 KB	8	2	640	1	112	2
FE	แผ่น 5.25 นิ้ว 180 KB	9	1	360	2	64	1
FF	แผ่น 5.25 นิ้ว 360 KB	9	2	720	2	112	2
F9	แผ่น 5.25 นิ้ว 1.2 MB	15	2	2400	7	224	1
	แผ่น 3.5 นิ้ว 720 KB	9	2	1440	3	112	2
	แผ่น 3.5 นิ้ว 1.44 MB	18	2	2880	9	224	1
F8	ฮาร์ดดิสค์ * 10 MB	17	4	20740	8	512	8
	ฮาร์ดดิสค์ * 20 MB	17	4	41752	41	512	4
	ฮาร์ดดิสค์ * 30 MB	17	4	62203	61	512	4
	ฮาร์ดดิสค์ * 238R, 32 MB	26	4	63726	63	512	4

*พารามิเตอร์ของฮาร์ดดิสค์จะไม่แน่นอนตายตัว อาจเปลี่ยนแปลงตามแต่การผลิตของแต่ละบริษัท

รูปที่ 8
รหัสประเภทของดิสค์
พร้อมพารามิเตอร์ที่สำคัญ

โปรแกรมเพื่อการบูตอยู่ที่ไบต์ 0, 1, 2 และไปต่อที่ไบต์หลังพารามิเตอร์ต่าง ๆ ที่แสดงในรูปที่ 7 ในตอนท้ายของบูตเรคอร์ดแสดงข้อความอธิบายความผิดพลาดที่เกิดจากแผ่นดิสค์ไม่มีไฟล์ระบบอยู่หรือดิสค์ใช้งานไม่ได้ และสุดท้ายจะจบด้วยข้อมูลลายเซ็นสองไบต์เป็น 55 AA

ไวรัสคอมพิวเตอร์หลายตัว เช่น (C) BRAIN มีการเปลี่ยนแปลงแก้ไขข้อมูลในบูตเรคอร์ดคือ ถ้าแผ่นดิสค์ติดไวรัสแล้ว ข้อมูลในบูตเรคอร์ดจะผิดไปจากเดิม ปรากฏเหตุการณ์ประหลาดที่ไม่ควรเกิด วิธีแก้ไขก็ต้องทำการฟอร์แมตแผ่นดิสค์นี้ใหม่

พื้นที่ของบูตเรคอร์ดเป็นส่วนสำคัญมาก เพราะดอสจะต้องเรียกใช้ด้วยเสมอ ถ้าผิวจานบริเวณเซกเตอร์แรกนี้เก็บข้อมูลไม่ได้ ก็หมายถึงแผ่นดิสค์นั้นใช้งานอีกต่อไปไม่ได้ ดังนั้นต้องระมัดระวังไม่บูตระบบซ้ำ ๆ ทันทีทันใดบ่อย ๆ

บูตเรคอร์ดในฮาร์ดดิสค์

ในกรณีของฮาร์ดดิสค์ วิธีจัดการโครงสร้างดิสค์จะแตกต่างออกไป โดยที่ฮาร์ดดิสค์มีพื้นที่เพื่อเก็บข้อมูลมาก จึงมีการแบ่งแยกเป็นพาร์ติชันได้รวมสี่พาร์ติชัน แต่ละพาร์ติชันก็อาจเป็นการใช้งานด้วยโปรแกรมควบคุมดำเนินงานที่ต่างกันได้ ซึ่งในส่วนนี้จะเป็นหน้าที่ของโปรแกรม FDISK ที่จะแบ่งกำหนดพื้นที่พาร์ติชันด้วยการระบุขนาดบอกตำแหน่งเริ่มต้นและจบท้ายของพาร์ติชัน แต่ละพาร์ติชันจะบรรจุโปรแกรมควบคุมดำเนินงานได้เพียงระบบเดียว โปรแกรม FDISK จะให้เราเลือกบอกพาร์ติชันที่แอคทีฟเพื่อเรียกไปสู่โปรแกรมควบคุมดำเนินงานที่บรรจุในพาร์ติชันนั้น รายละเอียดของโปรแกรม FDISK ได้เคยนำเสนอในมุมโอเอสไปแล้ว

การใช้งานทั่วไปในปัจจุบัน นิยมกำหนดให้พื้นที่ดิสค์เป็นส่วนของดอสทั้งหมด ถ้าขนาดความจุของดิสค์น้อยกว่า 32 เมกะไบต์ก็จะแบ่งมีเพียงพาร์ติชันเดียว แต่ถ้าขนาดความจุดิสค์มากกว่า 32 เมกะไบต์ เราสามารถแบ่งดิสค์ออกเป็นหลายพาร์ติชัน โดยพาร์ติชันแรกเป็นพื้นที่ดอสเต็ม 32 เมกะไบต์ และพาร์ติชันต่อ ๆ ไปเป็นส่วนเพิ่มเติมขยายจากพาร์ติชันแรกที่มีการระบุเป็นลอจิคัลไดรฟ์อื่น เช่น เป็นไดรฟ์ D หรือ E ต่อ ๆ ไป สำหรับในดอส 4.0 เราสามารถกำหนดพื้นที่แต่ละพาร์ติชันสูงกว่า 32 เมกะไบต์ได้แล้ว โดยจะได้สูงถึง 120 เมกะไบต์ทีเดียว

ข้อมูลต่าง ๆ ที่เรากำหนดขณะเรียกใช้โปรแกรม FDISK เพื่อกำหนดขนาดพาร์ติชันและเลือกพาร์ติชันใดที่แอคทีฟ จะเก็บอยู่ในเซกเตอร์แรกของฮาร์ดดิสค์เรียกว่าบูตเรคอร์ดหลัก (master boot record) คือจะอยู่ที่ไซเลนเดอร์ 0 หน้า 0 เซกเตอร์ 1 รูปที่ 9 เป็นตัวอย่างของบูตเรคอร์ดหลัก ซึ่งในตอนต้นของเซกเตอร์เป็นโปรแกรมเพื่อเรียกไปยังพาร์ติชันที่แอคทีฟตามด้วยข้อความบอกความผิดพลาด และตารางพาร์ติชันเริ่มจากตำแหน่งไบต์ที่ 446 เป็นต้นไป พาร์ติชันละ 16 ไบต์ ตัวอย่างตารางพาร์ติชันนี้เป็นข้อมูลของฮาร์ดดิสค์ขนาด 40 เมกะไบต์ความหนาแน่น 26 เซกเตอร์ต่อไซเลนเดอร์ที่แบ่งเป็นพาร์ติชันหลักและพาร์ติชันเพิ่มขยาย รายละเอียดของข้อมูลที่เก็บในตารางพาร์ติชันจะอธิบายในรูปที่ 10 โดยพิจารณาจากข้อมูลของรูปที่ 9 ข้างต้น

จากข้อมูลที่ปรากฏในตารางพาร์ติชัน ทำให้ระบบมองเห็นว่าพาร์ติชันใดแอคทีฟอยู่เพื่อให้โปรแกรมสั่งการทำงานไปยังพาร์ติชันนั้น เซกเตอร์แรกภายในพาร์ติชันนั้นก็จะบรรจุบูตเรคอร์ดของโปรแกรมควบคุมดำเนินงาน โดยมีโครงสร้างเป็นโปรแกรมสั้น ๆ เพื่อเรียกโหลดโปรแกรมควบคุมดำเนินงานลงสู่หน่วยความจำ ขณะเดียวกันจะมีส่วนที่เป็นพารามิเตอร์ไบออส ตัวอย่างของบูตเรคอร์ดในพาร์ติชันของดอสได้แสดงไว้ในรูปที่ 11

ไวรัสคอมพิวเตอร์ที่กำลังระบาดบางตัวได้เข้าไปแก้ไขข้อมูลในบูตเรคอร์ดหลักของฮาร์ดดิสค์ที่อยู่เหนือจากการใช้โปรแกรม FORMAT เพื่อการฟอร์แมตระดับสูง ดังนั้นเราต้องไปทำการฟอร์แมตฮาร์ดดิสค์ใหม่ตั้งแต่การฟอร์แมตระดับต่ำมาเลย

เราได้กล่าวไว้แล้วว่า ถ้าเซกเตอร์แรกที่เป็นพื้นที่ของบูตเรคอร์ดเกิดเสียเก็บข้อมูลไม่ได้ แผ่นดิสเกตต์นั้นจะใช้งานต่อไปไม่ได้ ในกรณีของฮาร์ดดิสค์จะแตกต่างออกไป เพราะเราสามารถทำการพาร์ติชันแบ่งพื้นที่เพื่อหลีกเลี่ยงไม่ให้เซกเตอร์เสียหรือไซเลนเดอร์ที่เสียอยู่ภายในพาร์ติชัน หรือถ้าจำเป็น

Side 0, Cylinder 0, Sector 1 ═══════ Partition Table format =

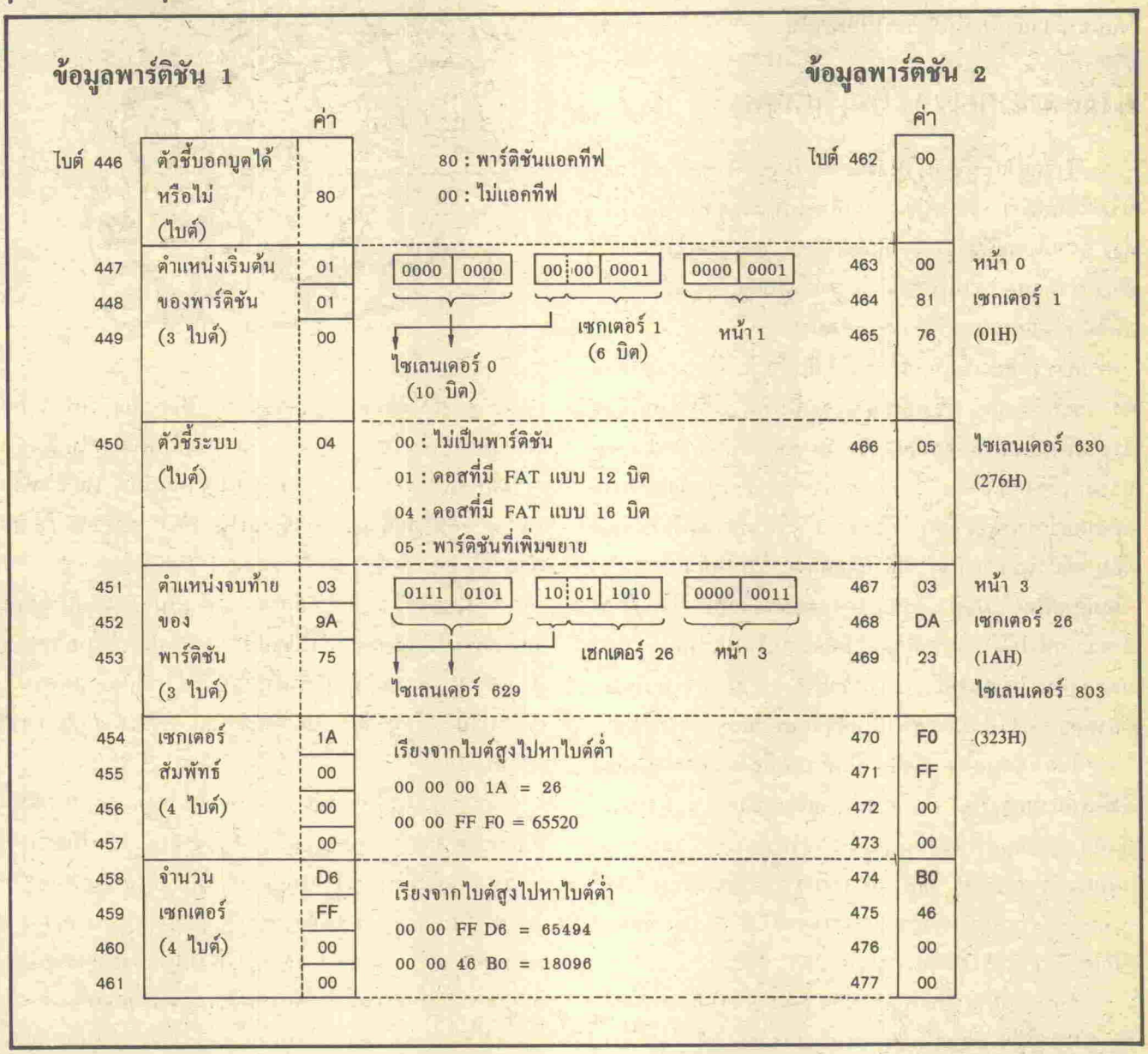
Partition Table Editor

System	Boot	Starting location Side	Cylinder	Sector	Ending location Side	Cylinder	Sector	Relative Sectors	Number of Sectors
DOS-16	Yes	1	0	1	3	629	26	26	65494
?	No	0	630	1	3	803	26	65520	18096
?	No	0	0	0	0	0	0	0	0
?	No	0	0	0	0	0	0	0	0

```
FA33C08E D0BC007C 8BF45007 501FFBFC BF0006B9 0001F2A5 EA1D0600 00BEBE07
B304803C 80740E80 3C00751C 83C610FE CB75EFCD 188B148B 4C028BEE 83C610FE
CB741A80 3C0074F4 BE8B06AC 3C00740B 56BB0700 B40ECD10 5EEBF0EB FEBF0500
BB007CB8 010257CD 135F730C 33C0CD13 4F75EDBE A306EBD3 BEC206BF FE7D813D
55AA75C7 8BF5EA00 7C000049 6E76616C 69642070 61727469 74696F6E 20746162
6C650045 72726F72 206C6F61 64696E67 206F7065 72617469 6E672073 79737465
6D004D69 7373696E 67206F70 65726174 696E6720 73797374 656D0000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00008001
01000403 9A751A00 0000D6FF 00000000 81760503 DA23F0FF 0000B046 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 000055AA
```

รูปที่ 9
ข้อมูลในบูตเรคอร์ดหลักของฮาร์ดดิสค์

รูปที่ 10 รายละเอียดข้อมูลที่เก็บเป็นตารางพาร์ติชัน




```
EB289049 424D2020 332E3300 02040100 020002D6 FFF84000 1A000400 1A000000
00000000 00000000 0000FA33 EDB8C007 8ED8C41E 1C008816 FD010AD2 7908891E
24008C06 26008EC5 8ED5BC00 7CFC1E36 C5367800 BF2A7CB9 0B00F3A4 1FC6062E
000FBF78 00B82A7C AB91ABFB 8A16FD01 CD13A010 0098F726 16000306 0E00E873
00E87900 BB000553 E8A0005F BE7101B9 0B0090F3 A6755783 C715B10B 9090F3A6
754C268B 471C998B 0E0B0003 C148F7F1 3D14007F 02B01496 A11100B1 04D3E8E8
3200FF36 2400C41E 6D01E830 00E85B00 2BF0760D E81D0052 F7260B00 03D85AEB
E95B8A2E 15008A16 FD01FF2E 6D01BE8B 01EB5490 01062400 112E2600 C3A11800
F6261A00 91A12400 8B162600 F7F1928B 0E1800F6 F12ACC91 FEC586E9 D0CED0CE
0AF186F2 87CA8A16 FD01C3BF 0500B801 02CD1372 03B001C3 80FC1174 F833C0CD
134F75EA BED501E8 1D00BEAB 01E81700 33C0CD16 36C70672 043412EA 0000FFFF
B40EBB07 00CD10AC 3C2475F4 C3000070 0049424D 42494F20 20434F4D 49424D44
4F532020 434F4D00 0000000A 0D4E6F6E 2D537973 74656D20 6469736B 206F7220
6469736B 20657272 6F72240A 0D526570 6C616365 20616E64 20737472 696B6520
616E7920 6B657920 7768656E 20726561 64790A0D 240A0D44 69736B20 626F6F74
20666169 6C757265 24436F70 722E2043 4F4D5041 51203139 38332D38 378055AA
```

รูปที่ 11
ข้อมูลในบูตเรคอร์ดของพาร์ติชันดอสเต็ม 32 เมกะไบต์

ต้องมีอยู่ในพาร์ติชันก็ต้องไม่ใช่เป็นบริเวณสำคัญของระบบให้เป็นบริเวณเพื่อเก็บข้อมูลไฟล์เท่านั้น

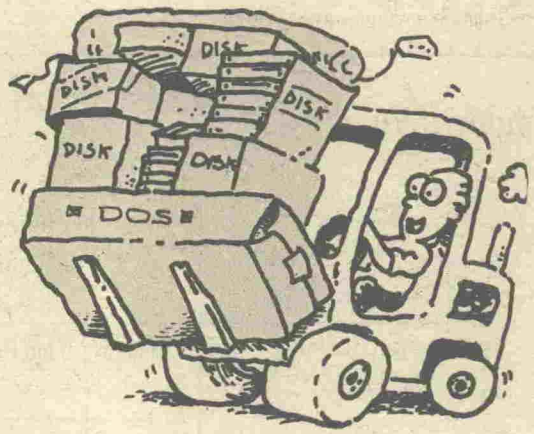
ผังจัดตำแหน่งของไฟล์ในดิสค์

ผังจัดตำแหน่งของไฟล์ในดิสค์ (File Allocation Table) หรือเรียกย่อ ๆ ว่า FAT จัดเป็นพื้นที่สำคัญที่จะบอกไฟล์ต่าง ๆ เก็บอยู่ที่ตำแหน่งไหนของดิสค์ และไฟล์นั้นจัดเก็บเรียงยาวติดต่อกันไปทีละเซกเตอร์ หรือแยกเก็บกระจายไปแทร็กหรือไซเลนเดอร์ต่าง ๆ อย่างไร

การจัดสรรเนื้อที่ดิสค์เพื่อใช้เก็บข้อมูล ดอสจะจัดให้คราวละคลัสเตอร์ ซึ่งคลัสเตอร์จะเป็นหน่วยที่ใช้ในการจัดวางไฟล์ในดิสค์ หนึ่งคลัสเตอร์จะเท่ากับกี่เซกเตอร์จะแล้วแต่ประเภทของดิสค์ เช่น แผ่นดิสค์ขนาด 360 กิโลไบต์จะมีขนาดคลัสเตอร์เท่ากับ 2 เซกเตอร์หรือ 1 กิโลไบต์ ดังนั้นไฟล์ที่มีข้อมูลเพียงไม่กี่ไบต์ก็จะใช้พื้นที่ดิสค์ถึง 1 กิโลไบต์ ทำให้เปลืองเนื้อที่ดิสค์โดยไม่ใช้ประโยชน์ การลดขนาดคลัสเตอร์ให้เล็กลงก็จะมีผลทำให้จำนวนคลัสเตอร์ที่ต้องเก็บในพื้นที่ที่เป็น FAT มากสูงขึ้น ดอสได้มีการคำนวณพิจารณาเลือกจำนวนและขนาดคลัสเตอร์ที่เหมาะสมไว้ให้ดังข้อมูลที่มีอยู่แล้วในรูปที่ 8

โดยที่ข้อมูลใน FAT เป็นข้อมูลที่สำคัญมาก ๆ ดอสจึงมีสำเนาของ FAT ไว้อีกหนึ่งชุดเป็นสองชุด ข้อมูล FAT ทั้งสองชุดจะอยู่เรียงต่อจากบูตเรคอร์ดทันที ตัวอย่างเช่น แผ่นดิสค์ขนาด 360 กิโลไบต์ จะมี FAT ชุดแรกที่เซกเตอร์ 2 และ 3 เรียงต่อมาจากบูตเรคอร์ดและ FAT ชุดที่สองที่เซกเตอร์ 4 และ 5 เรียงต่อจาก FAT ชุดแรก

ข้อมูลในตารางคลัสเตอร์ของ FAT แบ่งได้เป็น 2 กลุ่ม คือ FAT ที่มีจำนวนบิตเป็น 16 บิตและ 12 บิต FAT แบบ



12 บิตจะใช้กับดิสค์ขนาดเล็ก เช่น แผ่นฟลอปปี้และฮาร์ดดิสค์ 10 เมกะไบต์ FAT แบบ 16 บิตใช้กับดิสค์ขนาดใหญ่ เช่น ฮาร์ดดิสค์ตั้งแต่ 20 เมกะไบต์ขึ้นไป จำนวนบิต 12 บิตหรือ 16 บิต จะหมายถึงการแบ่งข้อมูลใน FAT คราวละ 12 บิตหรือ 16 บิต

คลัสเตอร์ 0 และคลัสเตอร์ 1 เป็นของระบบโดยจะแสดงรหัสประเภทดิสค์ (ดังรูปที่ 8) ที่เหลือจะเป็นเลขฐานสิบหก F ข้อมูลไฟล์ที่เก็บในดิสค์จะเริ่มจากคลัสเตอร์ที่ 2 เป็นต้นไป ข้อมูลที่เก็บในแต่ละคลัสเตอร์ทั้งสองกลุ่มแสดงในรูปที่ 12

รูปที่ 13 เป็นตัวอย่างข้อมูลจาก FAT แบบ 12 บิต ดิสค์ขนาด 360 กิโลไบต์ ซึ่งเราจะต้องนำข้อมูลของไบต์ต่าง ๆ มาเรียงกันแล้วอ่านคราวละ 12 บิตหรือ 3 หลักของเลขฐานสิบหก ดังในรูปที่ 14 ไฟล์แรกจะเริ่มที่คลัสเตอร์ 2 ข้อมูลในคลัสเตอร์ 2 นี้แสดงค่า 3 บอกว่าไฟล์นี้ยังไม่จบต้องต่อไปที่คลัสเตอร์ 3 คลัสเตอร์ 3 ก็ต่อไปคลัสเตอร์ 4 เรียงเช่นนี้ติดต่อกันจนถึงคลัสเตอร์ที่ 25 มีค่าเป็น FFF แสดงว่าคลัสเตอร์นี้

เลขฐานสิบหก FAT แบบ 12 บิต	เลขฐานสิบหก FAT แบบ 16 บิต	ความหมาย
000	0000	คลัสเตอร์ว่างไม่ได้ใช้
001	0001	คลัสเตอร์มีการใช้แต่ค่าตัวเลขผิดเพราะคลัสเตอร์จะเริ่มจากคลัสเตอร์สองเป็นต้นไป
002-FEF	0002-FFEF	คลัสเตอร์มีการใช้ ค่าตัวเลขจะแสดงคลัสเตอร์ที่ต่อจากคลัสเตอร์ปัจจุบัน
FF0-FF6	FFF0-FFF6	ค่าคลัสเตอร์ที่สงวนนำไปใช้ไม่ได้
FF7	FFF7	คลัสเตอร์เสียใช้งานไม่ได้
FF8-FFF	FFF8-FFFF	คลัสเตอร์สุดท้ายของไฟล์ไม่ต้องโยงต่อไปคลัสเตอร์อื่นอีกเป็นสัญลักษณ์บอกการจบของไฟล์

รูปที่ 12
ข้อมูลที่เก็บในคลัสเตอร์ของ FAT

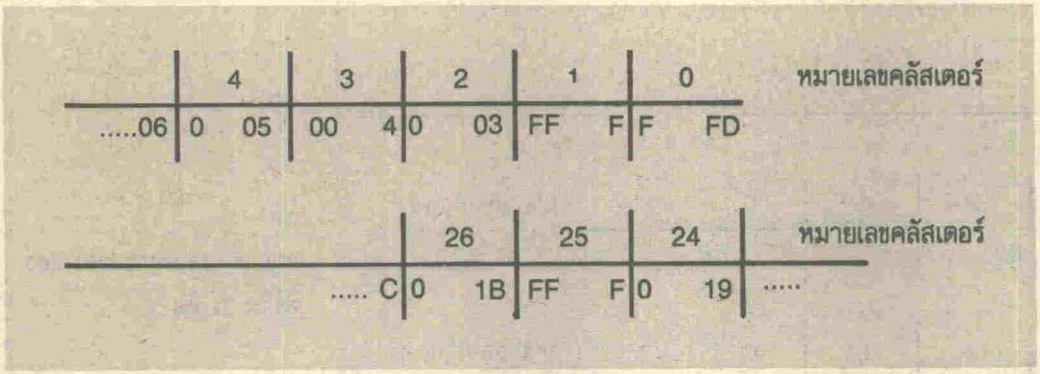
Sector 1 in 1st copy of FAT — Cluster 2, hex 0002

3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	<EOF>
27	28	29	30	31	32	33	34	35	36	37	38
39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	<EOF>	57	58	59	60	61	62
63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	<EOF>	82	83	84	85	86
87	88	89	90	91	92	93	94	95	96	97	98
99	100	101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146
147	148	149	150	151	152	153	154	155	156	157	158
159	160	161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180	181	182
183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	<EOF>	204	205	206
207	208	209	210	211	212	213	<EOF>	215	216	<EOF>	218
<EOF>	224	221	<EOF>	223	<EOF>	<EOF>	226	<EOF>	0	0	0

Sector 1 in 1st copy of FAT

```
FDFFFF03 40000560 00078000 09A0000B C0000DE0 000F0001 11200113 40011560
01178001 19F0FF1B C0011DE0 011F0002 21200223 40022560 02278002 29A0022B
C0022DE0 022F0003 31200333 40033560 0337F0FF 39A0033B C0033DE0 033F0004
41200443 40044560 04478004 49A0044B C0044DE0 044F0005 FF2F0553 40055560
05578005 59A0055B C0055DE0 055F0006 61200663 40066560 06678006 69A0066B
C0066DE0 066F0007 71200773 40077560 07778007 79A0077B C0077DE0 077F0008
81200883 40088560 08878008 89A0088B C0088DE0 088F0009 91200993 40099560
09978009 99A0099B C0099DE0 099F000A A1200AA3 400AA560 0AA7800A A9A00AAB
C00AADE0 0AAF000B B1200BB3 400BB560 0BB7800B B9A00BBB C00BBDE0 0BBF000C
C1200CC3 400CC560 0CC7800C C9A00CFF CF0CCDE0 0CCF000D D1200DD3 400DD5F0
FFD7800D FFAF0DFF 0F0EDDF0 FFDFF0FF FF2F0EFF 0F000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

รูปที่ 13
ตัวอย่างข้อมูล FAT
แบบ 12 บิต



รูปที่ 14
การอ่านข้อมูลตามหมายเลข
คลัสเตอร์ที่ใช้ทีละ 12 บิต

เป็นคลัสเตอร์สุดท้าย เราสามารถสรุปได้ว่า ไฟล์นี้ใช้พื้นที่ดิสค์รวม 24 คลัสเตอร์หรือ 24 กิโลไบต์ และใช้พื้นที่ดิสค์เรียงติดต่อกัน ไม่ต้องกระโดดข้ามไซเลนเดอร์อื่น

รูปที่ 15 เป็นตัวอย่างข้อมูลจาก FAT แบบ 16 บิตของฮาร์ดดิสค์ การวิเคราะห์ข้อมูลจะนับคราวละ 4 หลักของเลขฐานสิบหก ดังรูปที่ 16

การอ่านข้อมูลจะกระทำในลักษณะเหมือน FAT แบบ 12 บิต แต่แบ่งคราวละ 16 บิต จากตัวอย่างข้อมูลนี้ไฟล์ข้อมูลหลายไฟล์ไม่ได้เก็บเรียงติดต่อกัน มีการกระโดดข้ามแทร็กหรือไซเลนเดอร์

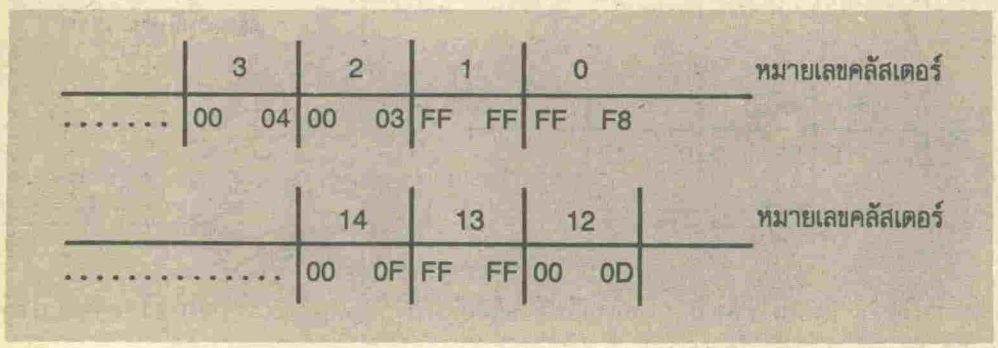
ถ้าหากไฟล์ใดมีการกระโดดข้ามไป ๆ มา ๆ ระหว่างไซเลนเดอร์และหัวอ่านเขียนมาก ๆ จะทำให้การเรียกค้นหรือเข้าถึงข้อมูลช้ากว่าปกติ ในอีกทางหนึ่งเราอาจใช้โปรแกรมยูทิลิตี้มาบีบอัดและจัดไฟล์ให้เรียงติดต่อกัน ซึ่งการแก้ไขนี้ก็คือการโยกย้ายตำแหน่งข้อมูลไฟล์ การจัดทำ FAT ขึ้นใหม่แทนที่ของเก่านั่นเอง

```
Sector 1 in 1st copy of FAT                                  Cluster 2, hex 0002

     3     4     5     6     7     8     9    10    11    12    13 <EOF>
    15    16    17    18    19    20    21    22    23    24    25    26
    27    28 <EOF> <EOF> <EOF>  1657    34    35   149    36    37    38
    39    40    43  2464  3693    44    45    46    51     0 <EOF> <EOF>
 <EOF>   145 <EOF>    54    55    56    57    58    59    60    61    62
    63    64    65    66    67    68 <EOF>    70    71    72    73    74
    75 <EOF>    77    78    79    80    81    82    83    84    85    86
    87    88    89    90    91 <EOF> <EOF>    94    95    96    97    98
    99   100   101   102   103   104   105   106 <EOF>   108   109   110
   111   112   113   114   115   116   117   118   119   120   121   122
 <EOF> <EOF>   125   126   127   128   129 <EOF>   131   132 <EOF> <EOF>
 <EOF>   136 <EOF>   138   139 <EOF>   141 <EOF> <EOF>   144   147   146
   155 <EOF> <EOF> <EOF>   175     0     0   154   156   161   157 <EOF>
   159   160   167   162   163   165 <EOF>   382 <EOF>   168   169   170
   180 <EOF>   173 <EOF> <EOF> <EOF>   177 <EOF>   179 <EOF>   182 <EOF>
   183   184   185   186   187   188   189   191 <EOF>   197 <EOF> <EOF>
 <EOF> <EOF> <EOF>   198   199   200   201   204   203   205   208   206
   207 <EOF>   209   210   211   212   216 <EOF> <EOF> <EOF>   250 <EOF>
   333   220   221   222   223   224 <EOF>   226   227 <EOF>   229   230
                .....
                .....
                .....

 Sector 1 in 1st copy of FAT
F8FFFFFF 03000400 05000600 07000800 09000A00 0B000C00 0D00FFFF 0F001000
11001200 13001400 15001600 17001800 19001A00 1B001C00 FFFFFFFF FFFF7906
22002300 95002400 25002600 27002800 2B00A009 6D0E2C00 2D002E00 33000000
FFFFFFFF FFFF9100 FFFF3600 37003800 39003A00 3B003C00 3D003E00 3F004000
41004200 43004400 FFFF4600 47004800 49004A00 4B00FFFF 4D004E00 4F005000
51005200 53005400 55005600 57005800 59005A00 5B00FFFF FFFF5E00 5F006000
61006200 63006400 65006600 67006800 69006A00 FFFF6C00 6D006E00 6F007000
71007200 73007400 75007600 77007800 79007A00 FFFFFFFF 7D007E00 7F008000
8100FFFF 83008400 FFFFFFFF FFFF8800 FFFF8A00 8B00FFFF 8D00FFFF FFFF9000
93009200 9B00FFFF FFFFFFFF AF000000 00009A00 9C00A100 9D00FFFF 9F00A000
A700A200 A300A500 FFFF7E01 FFFFA800 A900AA00 B400FFFF AD00FFFF FFFFFFFF
B100FFFF B300FFFF B600FFFF B700B800 B900BA00 BB00BC00 BD00BF00 FFFFC500
FFFFFFFF FFFFFFFF FFFFC600 C700C800 C900CC00 CB00CD00 D000CE00 CF00FFFF
D100D200 D300D400 D800FFFF FFFFFFFF FA00FFFF 4D01DC00 DD00DE00 DF00E000
FFFFE200 E300FFFF E500E600 E700E800 E900EA00 FFFFEC00 ED00EE00 F700F000
F100F200 F300F400 F500F600 FFFFF800 F900FFFF FFFF0000 00000000 00000000
                .....
                .....
                .....
```

รูปที่ 15
ตัวอย่างข้อมูล FAT
แบบ 16 บิต

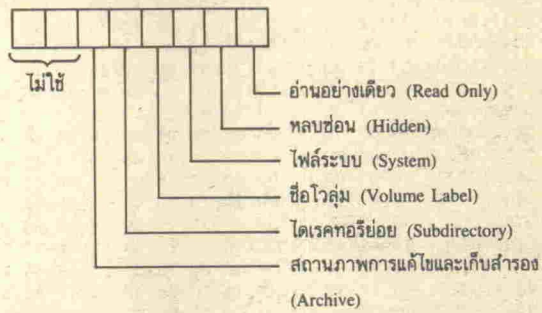
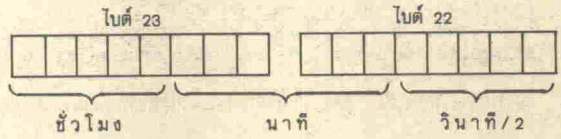
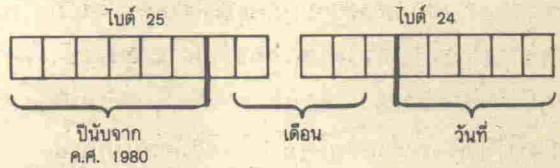


รูปที่ 16
การอ่านข้อมูลตามหมายเลขคลัสเตอร์
ที่ใช้ทีละ 16 บิต

ไดเรคทอรีราก (Root Directory)

พื้นที่ไดเรคทอรีรากจะเป็นพื้นที่ที่อยู่ต่อจากพื้นที่ FAT เพื่อเก็บข้อมูลรายการไฟล์ต่าง ๆ ที่บรรจุในไดเรคทอรีระดับราก โดยจะรวมชื่อโวลุ่มและชื่อไดเรคทอรีย่อยต่าง ๆ แต่ละรายการไฟล์ ชื่อโวลุ่ม และชื่อไดเรคทอรีย่อยจะใช้เนื้อที่รวม 32 ไบต์ดังรูปที่ 17 ดิสค์ประเภทต่าง ๆ จะกำหนดให้มีจำนวนรายการเพื่อระบุในไดเรคทอรีรากไม่เท่ากัน แผ่นดิสค์ขนาด 360 กิโลไบต์ จะเก็บรายการไฟล์ได้ 112 รายการ เมื่อคูณค่า

รูปที่ 17
รายการไดเรคทอรีของไฟล์

ไบต์ที่	เนื้อหา	คำอธิบาย
0-17 (8ไบต์)	ชื่อไฟล์	- มีได้ถึง 8 ตัวอักษร ถ้าชื่อไฟล์ยาวไม่ถึง 8 ตัวอักษรเนื้อที่ที่เหลือจะเป็นช่องว่าง - ถ้าไบต์แรกเป็นศูนย์ แสดงว่าส่วนที่เหลือไม่ได้ใช้ว่างอยู่ - ถ้าไบต์แรกเป็นจุดหรือข้อมูลเป็น 2E (เลขฐานสิบหก) และไบต์ต่อไปเป็นช่องว่าง จะแสดงเป็นไดเรคทอรีปัจจุบันใช้เป็นการอ้างอิงในการเรียกไดเรคทอรีย่อย - ถ้าสองไบต์แรกเป็นจุด แล้วตามด้วยช่องว่าง จะแสดงว่าเป็นไดเรคทอรีของพ่อแม่ที่อยู่เหนือไดเรคทอรีปัจจุบันใช้ในการอ้างอิงเรียกใช้ไดเรคทอรีย่อย - ถ้าไบต์แรกเป็น E5 (เลขฐานสิบหก) แสดงว่าไฟล์นั้นถูกสั่งลบออก โดยเปลี่ยนตัวอักษรตัวแรกของชื่อไฟล์เป็น E5 เท่านั้น ข้อมูลอื่นยังคงเดิม
8-10 (3 ไบต์)	นามสกุลไฟล์	- มีได้ 3 ตัวอักษร ถ้าใส่ยาวไม่ถึง 3 ตัวอักษร เนื้อที่ที่เหลือจะเป็นช่องว่าง
11 (1 ไบต์)	แอตทริบิ้วของไฟล์	- ข้อมูลแต่ละบิตจะมีความหมายเกี่ยวกับไฟล์ 
12-21 (10 ไบต์)	ไม่มีการใช้งาน	- สำรองไว้เพื่อใช้ในอนาคต
22-23 (2 ไบต์)	เวลา	- ไฟล์ที่มีการสร้างหรือดัดแปลงใหม่จะมีข้อมูลเวลาขณะนั้นเก็บไว้ด้วยแยกเป็น 2 ไบต์ มีความหมายดังนี้ 
24-25 (2 ไบต์)	วันที่	- ไฟล์ที่มีการสร้างหรือดัดแปลงแก้ไขใหม่ จะมีข้อมูลวันที่ขณะนั้นเก็บไว้เป็นสองไบต์ มีความหมายดังนี้ 
26-27 (2 ไบต์)	หมายเลขคลัสเตอร์แรกของไฟล์	- ข้อมูลของไฟล์ที่เก็บในดิสค์ จะชี้บอกตำแหน่งเก็บด้วยการบอกหมายเลขคลัสเตอร์แรกของไฟล์ จากนั้นจึงไปตรวจค้นใน FAT ต่อไป
28-31 (4 ไบต์)	ขนาดไฟล์	- ขนาดของไฟล์จะบอกเป็นจำนวนไบต์

```
                Absolute sector 0000127, System ROOT, Disk Abs Sec 0000127
0000(0000) 53 4B 20 20 20 20 20 20 20 20 20 10 00 00 00 00  SK          ▸
0016(0010) 00 00 00 00 00 00 91 01 21 00 34 00 00 00 00 00        æ☺! 4
0032(0020) 43 48 41 49 59 4F 4E 47 20 20 20 28 00 00 00 00  CHAIYONG    (
0048(0030) 00 00 00 00 00 00 00 53 7D 79 11 00 00 00 00 00         S}y◂
0064(0040) 57 50 35 31 20 20 20 20 20 20 20 10 00 00 00 00  WP51        ▸
0080(0050) 00 00 00 00 00 00 A0 02 21 00 2D 00 00 00 00 00        á☻! -
0096(0060) 31 32 33 20 20 20 20 20 20 20 20 10 00 00 00 00  123         ▸
0112(0070) 00 00 00 00 00 00 00 A2 A1 46 13 18 00 00 00 00         óíF‼↑
0128(0080) 54 50 35 30 20 20 20 20 20 20 20 10 00 00 00 00  TP50        ▸
0144(0090) 00 00 00 00 00 00 F5 7E 79 11 24 00 00 00 00 00        ⌡~y◂$
0160(00A0) 31 30 31 4D 41 43 52 4F 20 20 20 10 00 00 00 00  101MACRO    ▸
0176(00B0) 00 00 00 00 00 00 2C 00 21 00 31 00 00 00 00 00        , ! 1
0192(00C0) 48 47 20 20 20 20 20 20 20 20 20 10 00 00 00 00  HG          ▸
0208(00D0) 00 00 00 00 00 00 5B A0 6C 15 1B 00 00 00 00 00        [álş
0224(00E0) 44 42 41 53 45 20 20 20 20 20 20 10 00 00 00 00  DBASE       ▸
0240(00F0) 00 00 00 00 00 00 65 01 21 00 57 00 00 00 00 00        e☺! W
```

ก. รายการไฟล์และไดเรคทอรีในไดเรคทอรีราก

```
                Absolute sector 0000259, Clust 00027, Disk Abs Sec 0000259
0000(0000) 2E 20 20 20 20 20 20 20 20 20 20 10 00 00 00 00  .           ▸
0016(0010) 00 00 00 00 00 00 5B A0 6C 15 1B 00 00 00 00 00        [álş
0032(0020) 2E 2E 20 20 20 20 20 20 20 20 20 10 00 00 00 00  ..          ▸
0048(0030) 00 00 00 00 00 00 5B A0 6C 15 00 00 00 00 00 00        [álş
0064(0040) 53 59 4D 42 4F 4C 53 20 20 20 20 10 00 00 00 00  SYMBOLS      ▸
0080(0050) 00 00 00 00 00 00 05 B3 6C 15 42 00 00 00 00 00        ♣│lşB
0096(0060) 48 47 50 49 45 20 20 20 43 48 54 20 00 00 00 00  HGPIE    CHT
0112(0070) 00 00 00 00 00 00 BB 85 81 0F 6F 0B 56 04 00 00        ╗àü oV♦
0128(0080) 48 47 20 20 20 20 20 20 43 48 54 20 00 00 00 00  HG        CHT
0144(0090) 00 00 00 00 00 00 92 5B 8D 14 BF 00 BB 3C 00 00        Æ[
0160(00A0) 48 47 20 20 20 20 20 20 45 58 45 20 00 00 00 00  HG        EXE
0176(00B0) 00 00 00 00 00 00 15 99 BD 14 E2 0B E0 08 02 00        §öJ¶Γ●
0192(00C0) 48 47 44 50 20 20 20 20 45 58 45 20 00 00 00 00  HGDP      EXE
0208(00D0) 00 00 00 00 00 00 A0 78 BA 14 0F 3A A0 9C 03 00        áx‖¶:á£♥
0224(00E0) 48 47 32 33 5F 31 20 20 4F 56 4C 20 00 00 00 00  HG23_1    OVL
0240(00F0) 00 00 00 00 00 00 C6 8B BD 14 6A 0F 60 46 04 00        ├ï╜¶j`F♦
```

ข. รายการไฟล์และไดเรคทอรีในไดเรคทอรีย่อย HG

```
                Absolute sector 0000005, System ROOT, Disk Abs Sec 0000005
0256(0100) 43 42 4F 4F 54 20 20 20 54 58 54 20 00 00 00 00  CBOOT     TXT
0272(0110) 00 00 00 00 00 00 00 47 68 2C 14 51 00 9C 04 00 00        Gh,¶Q £♦
0288(0120) E5 46 41 54 20 20 20 20 42 41 4B 20 00 00 00 00  σFAT      BAK
0304(0130) 00 00 00 00 00 00 67 83 2C 14 61 00 0E 0B 00 00         gâ,¶a
0320(0140) 43 46 41 54 20 20 20 20 54 58 54 20 00 00 00 00  CFAT      TXT
0336(0150) 00 00 00 00 00 00 93 83 2C 14 64 00 10 0B 00 00        ôâ,¶d ▸
```

ค. รายการไฟล์ CFAT.BAK ถูกสั่งลบออก จึงแสดงรหัสตัวแรกของชื่อไฟล์เป็น E5

รูปที่ 18
ตัวอย่างรายการไฟล์
โวลุ่ม และชื่อไดเรคทอรี

112 กับ 32 ไบต์ และหารด้วย 512 ไบต์/เซกเตอร์จะเป็นเนื้อที่รวม 7 เซกเตอร์พอดี ในกรณีที่รายการไฟล์มีมากกว่าจำนวนรายการที่ใช้เก็บในไดเรคทอรีราก เราอาจต้องดัดแปลงสร้างเป็นไดเรคทอรีย่อย แล้วนำรายการไฟล์ไปเก็บในไดเรคทอรีย่อย ทำให้เราเพิ่มจำนวนไฟล์ในดิสค์ได้อีกมากมาย

รูปที่ 18 ก. เป็นตัวอย่างรายการไฟล์ โวลุ่มและชื่อไดเรคทอรีในไดเรคทอรีราก รูปที่ 18 ข. เป็นตัวอย่างรายการไฟล์ที่สร้างในไดเรคทอรีย่อย และรูปที่ 18 ค. เป็นการลบไฟล์ออกจากดิสค์ แต่จะยังคงรหัส E5 ที่ตำแหน่งอักษรตัวแรกของชื่อไฟล์อยู่ด้วย เราสามารถกู้ไฟล์คืนมาด้วยการเปลี่ยนอักษรตัวแรกเป็นชื่อไฟล์เดิมที่เคยมีอยู่

ถัดจากพื้นที่ไดเรคทอรีรากก็จะเป็นพื้นที่สำหรับเก็บข้อมูลไฟล์ทั้งหมด เซกเตอร์แรกของพื้นที่ข้อมูลก็จะตรงกับคลัสเตอร์ที่สองทันที

การอ่านและเขียนไฟล์

พื้นที่ตั้งแต่บูตเรคอร์ด FAT และไดเรคทอรีรากจะมีการทำงานที่สัมพันธ์กัน โดยบูตเรคอร์ดจะบอกรายละเอียด

ของดิสค์กับดอสก่อนว่าในระบบมี FAT และไดเรคทอรีอย่างไร ถ้าเราต้องการอ่านข้อมูลจากไฟล์ที่มีชื่อและนามสกุลใด ๆ ดอสจะนำชื่อและนามสกุลไฟล์นั้นไปตรวจสอบกับรายการในไดเรคทอรี ถ้าพบก็จะตรวจดูว่าไฟล์นั้นบรรจุที่คลัสเตอร์แรกอะไร นำหมายเลขคลัสเตอร์นั้นไปตรวจดูคลัสเตอร์ข้อมูลที่ต่อเนื่องไปยังคลัสเตอร์อื่นกับ FAT จนถึงคลัสเตอร์ที่บอกจบไฟล์ จากหมายเลขคลัสเตอร์ต่าง ๆ ก็สามารถโยงไปถึงข้อมูลที่เก็บในดิสค์

การเขียนบันทึกข้อมูลไฟล์จะดำเนินในทางกลับกันคือขั้นแรกดอสต้องตรวจดูว่า FAT มีคลัสเตอร์อะไรว่างอยู่ โดยเช็คจากคลัสเตอร์ 2 ไป ถ้าพบว่าว่างก็จะจัดรายก่ารไฟล์ในไดเรคทอรีด้วยการใส่คลัสเตอร์แรกของไฟล์ไว้พร้อมกับข้อมูลอื่น ๆ หลังจากนั้นดอสจะต้องจัดเรียง FAT ให้รับกับขนาดของไฟล์ ถ้าขนาดไฟล์มากกว่าหนึ่งคลัสเตอร์ ดอสก็ต้องตรวจว่าคลัสเตอร์ถัดไปว่างหรือเสียหรือเปล่า เพื่อนำหมายเลขคลัสเตอร์ถัดไปมาใส่ในคลัสเตอร์ปัจจุบันเรียงไปจนครบขนาดของไฟล์ และคลัสเตอร์ท้ายใส่ข้อมูลบอกว่าจบคลัสเตอร์

เนื่องจากไฟล์ต่าง ๆ ที่เก็บในดิสค์มีการลบออกอยู่สม่ำเสมอ ทำให้เกิดคลัสเตอร์ว่างกระจัดกระจายอยู่ทั่วพื้นที่ การบรรจุไฟล์ใหม่ลงไปจึงเกิดการกระจายไม่ปะติดปะต่อเป็นกลุ่มเดียวกันได้เสมอ ผู้ใช้งานเฉพาะฮาร์ดดิสค์อาจมีการอัดบีบไฟล์ต่าง ๆ ให้ติดต่อกันและให้ข้อมูลไฟล์อยู่ในคลัสเตอร์อยู่เรียงติดกันด้วยโปรแกรมยูทิลิตี้ เช่น PC Tools DELUXE เป็นต้น

การใช้คำสั่ง FORMAT

ดอสจะใช้คำสั่ง FORMAT ในการเตรียมแผ่นดิสค์หรือฮาร์ดดิสค์เพื่อนำมาใช้ในระบบ โปรแกรม FORMAT นี้จะช่วยตรวจสอบดิสค์และกันส่วนที่เสียออกไป รูปแบบคำสั่ง FORMAT จะเป็นดังนี้

FORMAT d: [/S] [/V] [/1] [/8] [/4] [/B] [/T : tt] [/N : ss] [/F : nn]

โดย d : เป็นชื่อดิสค์ไดรฟ์ที่มีแผ่นดิสค์หรือฮาร์ดดิสค์เพื่อการฟอร์แมตและมีสวิตช์ให้เลือกใช้หลายตัวดังนี้

/S เลือกให้คัดลอกไฟล์ระบบ IBMDOS.COM, IBMBIO.COM และ COMMAND. COM ไปสู่ดิสค์เพื่อให้ดิสค์นั้นใช้บูตระบบได้

/1 เลือกฟอร์แมตดิสค์ใช้เฉพาะหน้าแรกหน้าเดียว แทนที่จะฟอร์แมตแผ่นดิสค์สองหน้า

/8 เลือกฟอร์แมตแผ่นดิสค์เป็น 8 เซกเตอร์ต่อแทร็ก แทนที่จะเป็น 9 หรือ 15 เซกเตอร์ต่อแทร็ก

/V เลือกฟอร์แมตแผ่นดิสค์ด้วยการระบุชื่อโวลุ่มความยาวสูงถึง 11 ตัวอักษร ในดอส 4.0 สามารถเติมชื่อต่อท้ายสวิตช์ได้ทันทีเป็น /V : name

/4 เลือกฟอร์แมตแผ่นดิสค์ความหนาแน่นสูงให้จุข้อมูลเพียง 360 กิโลไบต์

/B เลือกฟอร์แมตแผ่นดิสค์โดยกันพื้นที่สำหรับบรรจุไฟล์ระบบที่หลบซ่อนของ IBMBIO.COM และ IBMDOS.COM ซึ่งเราสามารถบรรจุไฟล์ทั้งสองไปลงช่องว่างที่เว้นไว้ด้วยคำสั่ง SYS ของดอสเวอร์ชันเดียวกัน

/T : tt เลือกฟอร์แมตแผ่นดิสค์ให้มีจำนวนแทร็กต่อหน้าเท่ากับ tt ตามปกติค่าที่ใช้เป็น /T : 80 มี 80 แทร็ก และ /T : 40 ฟอร์แมตให้มี 40 แทร็ก

/N : ss เลือกฟอร์แมตดิสค์ให้มีจำนวนเซกเตอร์ต่อแทร็กเท่ากับ ss ตามปกติค่าที่ใช้เป็น /N : 9 และ /N : 8

/F : nn เลือกฟอร์แมตดิสค์มีความจุต่ำกว่าความสามารถของไดรฟ์เป็นสวิตช์ใหม่เข้ามาในดอสเวอร์ชัน 4 เพื่อกำหนดความจุเป็นตัวเลขแทนที่การใช้ /T และ /N ที่ไม่สะดวกนัก ค่าที่นิยมใช้เป็น 160, 180, 320, 360, 720, 1200 และ 1440 ตามความจุมาตรฐานของแผ่นดิสเกตต์

สวิตช์ทุกตัวที่กล่าวมาแล้วไม่ได้ใช้ได้กับแผ่นดิสค์ทุกขนาด ตารางในรูปที่ 19 เป็นสวิตช์ที่ใช้กับดิสค์ประเภทต่าง ๆ โดยสวิตช์ /N และ /T จะแทนด้วยสวิตช์ /F : nn สวิตช์ที่กล่าวในตาราง บางครั้งจะใช้ร่วมกันได้และใช้ร่วมกันไม่ได้ ที่ใช้ร่วมกันไม่ได้ได้แก่

– /B ร่วมพร้อมกับ /S ไม่ได้

รูปที่ 19 สวิตช์ที่ใช้กับดิสค์ประเภทต่าง ๆ

ประเภทดิสค์ (ความจุ)	สวิตช์
160/180 กิโลไบต์	/S, /V : name, /1, /8, /B, /F : nn, /4
320/360 กิโลไบต์	/S, /V : name, /1, /8, /B, /F : nn, /4
720/1440 กิโลไบต์	/S, /V : name, /B, /F : nn
1.2 เมกะไบต์	/S, /V : name, /B, /F : nn
ฮาร์ดดิสค์	/S, /V : name, /B

– /N หรือ /T ร่วมกับแผ่น 320/360 กิโลไบต์หรือฮาร์ดดิสค์ไม่ได้

– /1, /4, /8 ร่วมกับฮาร์ดดิสค์ไม่ได้

ตัวอย่างการใช้งานคำสั่ง FORMAT

1. FORMAT A : <ENTER>

การฟอร์แมตแผ่นดิสค์ในช่องไดรฟ์ A ตามขนาดความจุของไดรฟ์ เมื่อทำการฟอร์แมตสำเร็จ แผ่นดิสค์นั้นจะมีการสร้างพื้นที่ 3 ส่วนคือ บูตเรคอร์ด FAT และไดเรคทอรีรากเกิดขึ้นทันที ตัวอย่างของแผ่นดิสค์ขนาด 360 กิโลไบต์หรือ 720 เซกเตอร์ จะถูกใช้เป็นบูตเรคอร์ด 1 เซกเตอร์ FAT ที่มีสองชุดเป็น 4 เซกเตอร์ และไดเรคทอรีรากเป็น 7 เซกเตอร์ ดังนั้นจะเหลือพื้นที่ว่างเพียง 708 เซกเตอร์ หรือ 354 กิโลไบต์เท่านั้นเอง

2. FORMAT A : /S <ENTER>

เหมือนการฟอร์แมตแบบในข้อ 1 เพียงแต่มีการคัดลอกโปรแกรมระบบ IBMBIO.COM IBMDOS.COM และ COMMAND.COM ไปสู่แผ่นดิสค์นั้น โดยเก็บไฟล์ทั้งสามในพื้นที่ข้อมูลต่อท้าย พื้นที่ไดเรคทอรีราก พื้นที่ไฟล์ IBMBIO.COM จะต้องต่อท้ายไดเรคทอรีราก แล้วจึงต่อด้วยไฟล์ IBMDOS.COM ทันที ไฟล์ COMMAND.COM จะเก็บที่ไหนไม่สำคัญ เมื่อมีการคัดลอกไฟล์ไปเก็บก็จะมีการสร้างรายการไดเรคทอรีของไฟล์และการแก้ไข FAT เพื่อแสดงตำแหน่งของไฟล์ในดิสค์

3. FORMAT A : /B <ENTER>

เหมือนการฟอร์แมตแบบในข้อ 1 แต่มีการเว้นพื้นที่เพื่อคัดลอก IBMBIO.COM และ IBMDOS.COM มาลงภายหลังด้วยคำสั่ง SYS ของดอสเวอร์ชันเดียวกัน ถ้ามีการสร้างไฟล์หรือคัดลอกไฟล์มาสู่ดิสค์ ไฟล์นั้นจะไม่มาใช้พื้นที่สงวนไว้ แต่จะใช้พื้นที่ที่จัดเลยออกไป

4. FORMAT A : /V : DISK1 <ENTER>

การฟอร์แมตดิสค์เหมือนข้อ 1 แต่เพิ่มการกำกับชื่อโวลุ่มเป็น DISK1 ลงไปด้วย ถ้าไม่มีชื่อกำกับต่อท้ายสวิตช์โปรแกรมจะถามให้เราป้อนชื่อภายหลังแทน

5. FORMAT A : /F : 720 <ENTER>

การฟอร์แมตแผ่นดิสค์ขนาด 3.5 นิ้วให้มีขนาดความจุ 720 กิโลไบต์ลงช่องไดรฟ์แบบ 1.44 เมกะไบต์

6. FORMAT A : /4 <ENTER>

การฟอร์แมตแผ่นดิสค์ขนาด 5.25 นิ้วให้มีความจุ 360 กิโลไบต์แต่ในช่องไดรฟ์แบบ 1.2 เมกะไบต์

สรุป

มุมโอเอสได้กล่าวถึงโครงสร้างและเทคโนโลยีของดิสค์มาหลายฉบับติดต่อกัน ในฉบับนี้ถือเป็นฉบับจบตอนที่กล่าวถึงการฟอร์แมตดิสค์ในระดับสูง ด้วยการใช้คำสั่ง FORMAT ของดอส โปรแกรม FORMAT จะดำเนินการจัดเตรียมการเก็บข้อมูลลงสู่ดิสค์ ด้วยการสร้างพื้นที่สำคัญของดิสค์แบ่งเป็นบูตเรคอร์ด ผังการจัดวางตำแหน่งไฟล์ในดิสค์ (FAT) และรายการไฟล์ในไดเรคทอรีราก

ข้อมูลที่กล่าวในฉบับนี้ค่อนข้างละเอียดและสมบูรณ์ เชื่อว่าจะเป็นประโยชน์ต่อผู้อ่าน ทั้งที่เพิ่งเข้าสู่วงการคอมพิวเตอร์หรืออยู่ในวงการมาพอสมควรแล้ว เพราะรายละเอียดในหลาย ๆ ส่วนจะช่วยเราดูแลรักษาข้อมูลในดิสค์ได้ดียิ่งขึ้น ถ้าเราเรียนรู้และเข้าใจวิธีการจัดเก็บข้อมูลในตอนนี้แล้ว จะช่วยให้เราเข้าใจเรื่องไวรัสคอมพิวเตอร์ที่กำลังระบาดได้ดียิ่งขึ้น ก็ขอแนะนำผู้อ่านค้นบทความด้านไวรัสคอมพิวเตอร์ในวารสารไมโครฉบับก่อน ๆ มาศึกษา แล้วผู้อ่านจะเข้าใจเรื่องไวรัสได้ง่ายขึ้น